Computing Reports

Chris Pinkenburg (BNL)
Jin Huang (BNL)

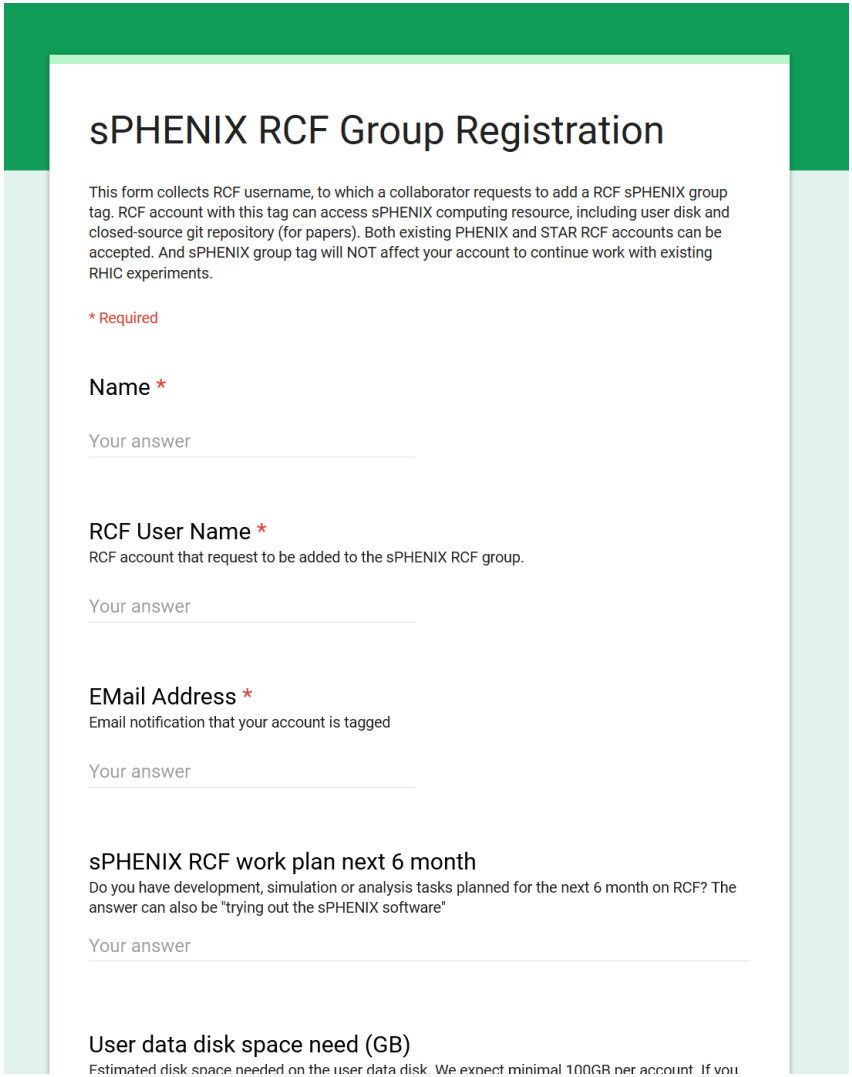
Computing News

- Registration
- sPHENIX Sim
- Test beam

Sign up for using sPHENIX RCF resource

- Add sPHENIX RCF group tag to your existing RCF account
  - For write use sPHENIX user data disk
    Expecting ~100GB/user. If need large space (10TB or large), talk to us for specific solution.
  - Write access to close-sourced Git repository (under test)
  - For resource expanding in the near future
- Sign up here:

http://goo.gl/forms/5SzP94oiHfq1dB802



sPHENIX RCF Group Registration

This form collects RCF username, to which a collaborator requests to add a RCF sPHENIX group tag. RCF account with this tag can access sPHENIX computing resource, including user disk and closed-source git repository (for papers). Both existing PHENIX and STAR RCF accounts can be accepted. And sPHENIX group tag will NOT affect your account to continue work with existing RHIC experiments.

* Required

Name *

Your answer

RCF User Name *

RCF account that request to be added to the sPHENIX RCF group.

Your answer

EMail Address *

Email notification that your account is tagged

Your answer

sPHENIX RCF work plan next 6 month

Do you have development, simulation or analysis tasks planned for the next 6 month on RCF? The answer can also be "trying out the sPHENIX software"

Your answer

User data disk space need (GB)

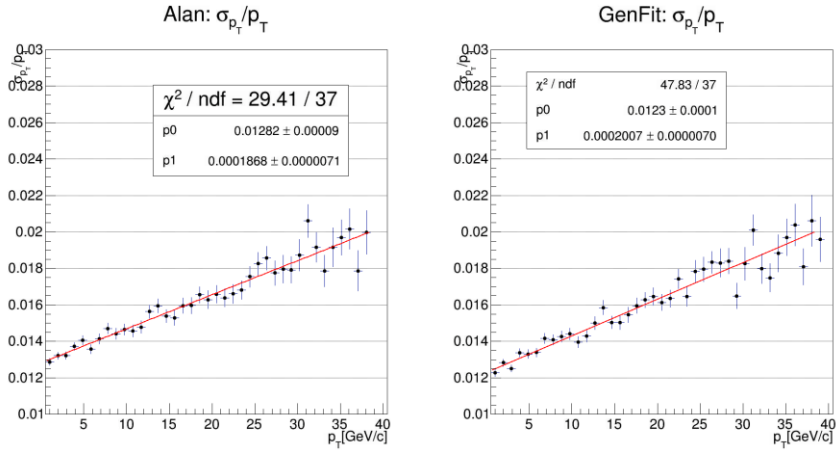
Estimated disk space needed on the user data disk. We expect minimal 100GB per account. If you

Computing news: Recent Developments

Simulation/software meetings: https://indico.bnl.gov/categoryDisplay.py?categId=88

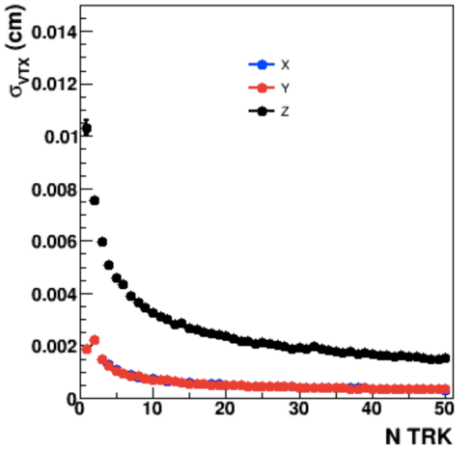
Adopting GenFit Kalman filter package

(Haiwang Y., Chris P., Mike M.)



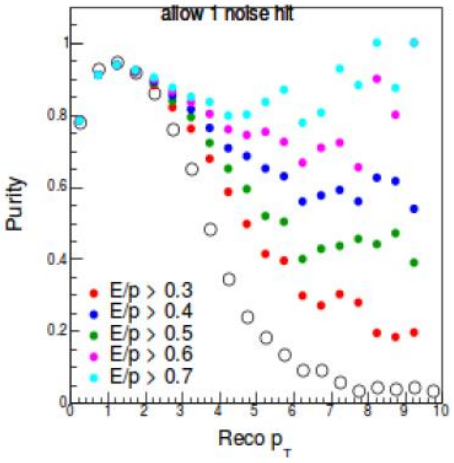
RAVE vertex finding/fitting

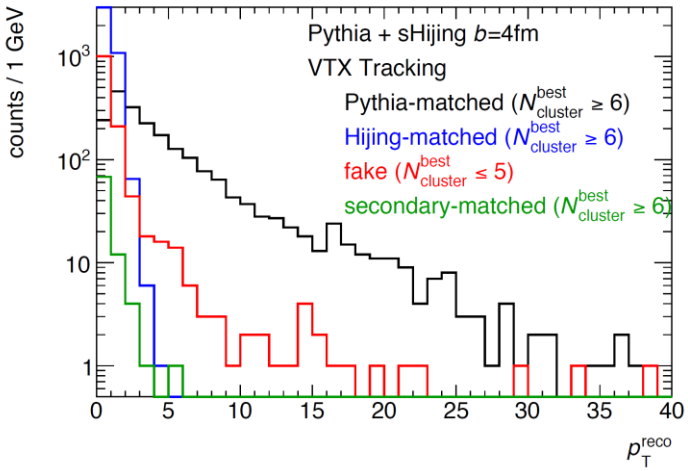
(Sanghoon L., Haiwang Y.)



Track purity in G4 tracker (VTX) + fast calorimeter sim.

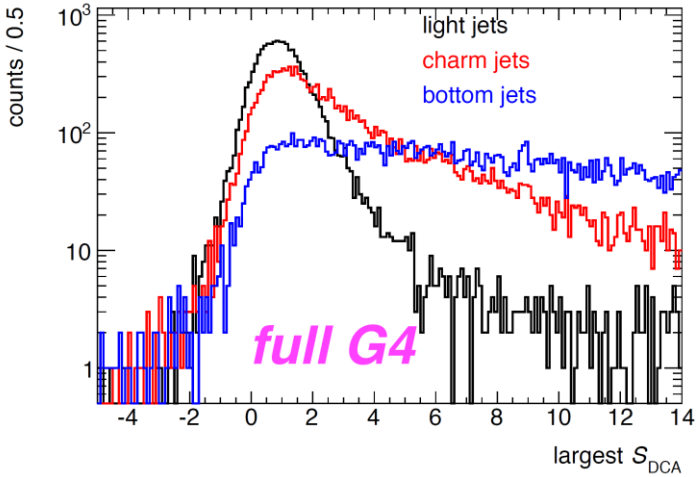
Inclusive track (Ron B./Kurt H.) , Track in jets (Dennis P.)





DCA-counting b-jet tagging in G4

(Dennis P.)

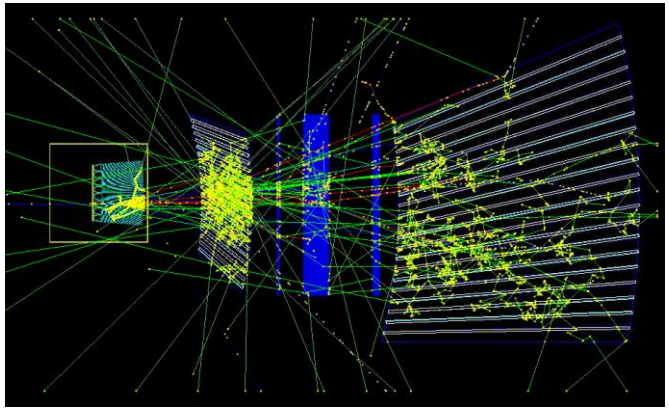


Computing news: Test beam in action

- Simulation
  (Chris P./Murad S./Jin H.)
  - One macro for everything: https://github.com/sPHENIX-Collaboration/macros/blob/master/macros/prototype2/Fun4All_G4_Prototype2.C
  - Quite comprehensive: Detailed geant4/Scintillation model/Photon fluctuation/digitization simulation
- Reconstruction
  (Abhisek S./Martin P./Jin H.)
  - Production in one macro: unpack and calibrate all channels/run info to DST
    https://github.com/sPHENIX-Collaboration/macros/blob/master/macros/prototype2/Fun4All_TestBeam.C
  - Frequently reproduce all 800+ runs with newest code-base and calibration
  - Read more:
    https://wiki.bnl.gov/sPHENIX/index.php/T-1044#Online_Production
- Analysis
  (Vera L./Abhisek S./Mike S./Ron B./Megan C./Jin H./John H./Ed K. and many others)
  - See analysis note writing in action:
    https://wiki.bnl.gov/sPHENIX/index.php/T-1044#Run_info

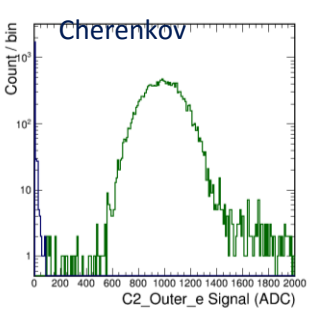
Simulation: event display

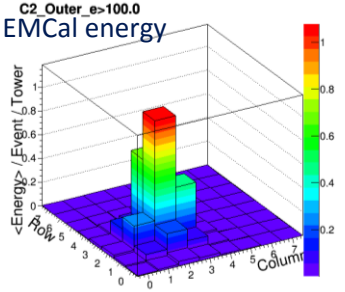
https://github.com/sPHENIX-Collaboration/macros/pull/22



Production: QA plots

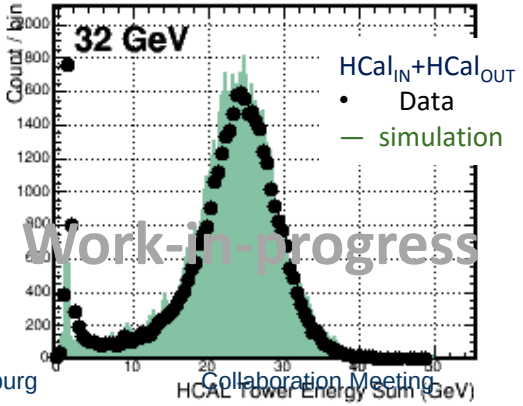
https://wiki.bnl.gov/sPHENIX/index.php/T-1044/joint_data_good_run_note





Analysis: data VS sim.

https://wiki.bnl.gov/sPHENIX/index.php/T-1044/HCAL_good_run_note



Computing resource (as of today)

CPU Resource

- sPHENIX is 64bit code to allow memory > 4GB (hijing with full calorimeter showers and truth info needs about 12GB)
- Computing done on PHENIX machines
  - STAR does not allow large memory jobs (>1.5GB?) which is not enough for even single particle sims,
  - PHENIX allows 3.8GB, STAR users can run on PHENIX nodes via general queue and tweaking requirements
- Special queue (high-memory) from up to 20GB jobs with limited number of slots so our 2GB/core machines survive
- Sufficient for what we are doing
  - Expect reduction in resource once PHENIX data production kicks in
- Exploration in open-science-grid
  - Possible additional avenue of few-thousand high memory slots for regular Geant4 full detector production.
  - Martin had a demonstration project with large Fun4All MB Pythia-6 production (for HF MB background study)
  - More work needed to further explore

Disk space

- 200TB /sphenix/user
  - That's the place for your files
  - Writable by sphenix group tag (sign-up first slide)
  - No quotas, we rely on your cooperation and good citizenship.
- 200TB /sphenix/sim/sim01
  - For central simulation production output
- 10TB /sphenix/data/data01
  - For test-beam data/production
- All on loan from PHENIX (should be safe, no immediate need by PHENIX foreseen)
- Hints:
  - PHENIX disks are still being re-organized/consolidated, please use softlinks /sphenix/user not mountpoints /gpfs02/ which are subject to change

Software status

https://wiki.bnl.gov/sPHENIX/index.php/Software

- And where could new collaborator contributes
- Also close to critical path of topical groups

Software framework

- Based on PHENIX software framework, a.k.a. Fun4All
  - Naturally supports pause analysis at any reconstruction stage (e.g. simulation/reconstruction/analysis or any sub steps), store intermediate data to file (a.k.a. DST file or PHENIX formatted ROOT file), and resume in another Fun4All reconstruction cycle
  - Naturally supports embedding, e.g. single particle in to A+A or Pythia8 p+p jet into A+A
  - Naturally supports event filtering, e.g. trigger on generator level (e.g. **PHPy8JetTrigger**) or reco level (easy to write when needed)
  - During event processing, reco data available in memory (**PHCompositeNode**) for user to write a module to analyze on-the-fly or choose to save relevant parts to user-defined Ntuple
  - Event mixing via analysis code
- Built-in Geant4 support
  - Constructs detector in Geant4 via C++ (Expert built and maintained)
  - Calls Geant4 track input primary particles, Record digested Geant4 hits
  - Deep truth ancestry tracing tool throughout analysis chain
  - Common macro run the simulation and standard analysis chain
- Easy access for user modules in analysis https://wiki.bnl.gov/sPHENIX/index.php/Example_of_using_DST_nodes
- Read more: Introduction to Fun4All, Running sPHENIX simulation
- Need:
  - Automatic QA for production and pull requests
  - A kick-ass reconstruction event display

Event generator

- Standard inputs: HEPMC format
  - Fun4AllHepMCInputManager -> HepMCNodeReader
- Ready-to-use generators
  - Pythia8 for p+p: PHPythia8
  - Pythia6 for p+p/e+p: PHPythia6
  - Hijing for p+A, A+A
    - option for after-burner of flow: flowAfterburner
  - Some home-brewed format of EIC Pythia6 input: ReadEICFiles
  - Of-course single particle generator for testing
- In development
  - For your study, welcome to contribute event generator trigger modules

Tracking Simulation

- Ready:
  - Cylinder shaped silicon tracker available
  - TPC + digitization ()
  - Ganging readout strips
  - Randomized dead channels
- Standard design options in standard macro: https://github.com/sPHENIX-Collaboration/macros/tree/master/macros/g4simulations
  1. Default: PHENIX VTX + RIKEN new strip layers in MIE: G4_Svtx.C
  2. PHENIX VTX + new TPC: G4_Svtx_pixels+tpc.C
  3. Maps inner pixel + RIKEN new strip layers: G4_Svtx_maps+strips.C
  4. Maps inner pixel + TPC: G4_Svtx_maps+tpc.C
  5. Full Maps (variation of ITS): G4_Svtx_ITS.C
- In development
  - Ladder based silicon tracker geometry (Tony Frawley, Kun Liu, and many more)
  - Continue develop for tracking options (tracking group)

Tracking reconstruction

- Ready
  - Patter recognition: hough transform based helical pattern reco in uniform field (PHG4HoughTransform)
  - Ghost/Fake rejection (PHG4TrackGhostRejection)
  - Kalman filter for cylindrical tracker (PHG4HoughTransform)
  - Track extrapolation (homogeneous field, PHG4SvtxTrackProjection)
  - Vertex finder via global minimal (VertexFinder)
  - Performance evaluator (SvtxEvaluator)
- On-going:
  - TPC reco speed tuning (Alan, Mike, Sourave)
  - Generic Kalman filter (Close to completion by Haiwang Yu)
  - Better handle of fake rejection (e.g. using calorimetry matching by Kurt Hill and Ron Belmont)
  - Quantification and comparison of silicon tracker options (tracking group)
  - RAVE secondary vertex finder (Sanghoon Lim and Haiwang Yu)
- Wish list
  - Generic pattern recognition including forward trackers (OLYMPUS tree-search?)

Calorimetry Simulation

- Ready
  - Detailed EM calorimeter sim based on UCLA SPACAL prototype and current sPHENIX engineering design of enclosure (PHG4SpacalSubsystem)
  - Detailed Hadron calorimeter based current sPHENIX engineering design (PHG4InnerHcalSubsystem and PHG4OuterHcalSubsystem)
  - Tower scheme with geometry description
  - Shower truth compression and association https://github.com/sPHENIX-Collaboration/coresoftware/pull/101
- On-going:
  - Fast calorimetry simulations (Ready in preview, tuning. Kurt Hill, Ron Belmont)
  - Tuning shower/digitization models: light collection, variation, hadron interaction model (calorimeter groups, test beam)
  - Calibration scheme (EMCal via pi0 by Vera Loggins)

Calorimetry reconstruction

- Clusterizer
  - Ready: a toy graph Clusterizer that connect all neighboring non-zero suppressed towers
  - Missing:
    Realistic Clusterizer that support non-spherical shower in full background
    - Especially important for γ-jet topical study, pi0 rejection and calibration
- Track – calorimeter association
  - Track projection based Clusterizer (ready)
  - Likelihood macro tool for electron ID (ready). Need to formulate a standardized module

Jet tools

- Baseline jet reco (JetReco)
  - Input: truth, track, tower, cluster
  - Algorithm: FastJet-based Clusterizer, i.e. Anti-kT, etc.
  - Output: Reconstructed jets, support deep truth association
- Exploratory:
  - CMS-style flow jet: PHFlowJetMaker by Javier Orjuela Koop
- Need
  - B-jet tagging (See HF Jet TG talk)
  - UE subtraction (coded in PHENIX software, need to be migrate over to sPHENIX and conform with code standard)
  - Formalize fake rejection, some quick form in PHENIX code base. Need to improve/port over to sPHENIX

Summary

- The good news is that so far we have sufficient resources
  - The model to run larger sims centrally seems to work
  - People to analyze large calorimeter simulation without having to learn how to runs them
  - We are running on the PHENIX part of RCF
- Framework is like LEGO – you get the building blocks and then you create something
- Many progress made recently on software development. Simulation and software meeting.
- Many opportunities to contribute
- Weekly updates: https://wiki.bnl.gov/sPHENIX/index.php/Software